一种基于群体智能的自组织重叠社团结构分析算法 *

孙韩林[a, b]，马素刚[a, b]，王忠民[a, b]

(西安邮电大学 a. 计算机学院; b. 陕西省网络数据智能处理重点实验室，西安 710121)

摘 要：社团结构分析是复杂网络研究的一项重要内容。基于群体智能思想提出了一种自组织的重叠社团结构分析算法 SO^2CSA^2。基本思想是：把网络视为一个群体，网络节点是其中的一个个具有简单智能的个体，每个个体依据定义的社团连接分数自主决定要加入的社团（可同时加入多个社团）。首先在网络中寻找一组 K-派系作为初始社团结构；在此基础上，所有个体迭代地选择其社团归属，最终整个网络的社团结构将逐渐生长出来；最后对获得的社团结构进行后处理，即调整少量节点的社团归属，以提高其质量。在一组合成网络和现实世界网络上的实验表明，SO^2CSA^2 发现的社团结构的质量比两种对比算法（SLPA 和 OSLOM）更好，尤其是在网络中重叠节点较多或节点重叠度较大的情况下，社团结构质量的提升更为明显。

关键词：重叠社团结构；社团检测；社团结构分析；复杂网络；群体智能

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.11.0733

Self-organizing overlapping community structure analysis algorithm based on swarm intelligence

Sun Hanlin[a, b], Ma Sugang[a, b], Wang Zhongmin[a, b]

(*a. School of Computer Science & Technology*, *b. Shaanxi Key Laboratory of Network Data Intelligent Processing*, *Xi'an University of Posts & Telecommunications*, *Xi'an* 710121, *China*)

Abstract: Community structure analysis is a critical task in examining a complex network. This paper presented a self-organizing overlapping community structure analysis algorithm (SO^2CSA^2) based on the swarm intelligence theory. The basic idea behind the algorithm was that it treats an analyzed network as a swarm intelligence system, of which each node was an individual with simple intelligence. Each individual independently decides to which community it joined based on a defined metric named connection score. An individual could join to multiple communities simultaneously. At first, the algorithm found a set of K-cliques from the analyzed network as the initial community structure. Then, each individual in the system acted iteratively to join into or leave from communities, and an optimal community structure of the whole network could develop and eventually emerge from the initial community structure. Finally, to improve the quality of the obtained community structure, a post process adjusted community assignments of a small number of individuals. Experimental evaluation on a number of synthesized networks and real-world networks indicates that the quality of community structures discovered by SO^2CSA^2 is better than those of two compared algorithms, SLPA and OSLOM, especially on networks with a large number of overlapping nodes or on networks with overlapping nodes of which overlapping degrees are high.

Key words: overlapping community structure; community detection; community structure analysis; complex network; swarm intelligence

0 引言

人类的生活和生产活动越来越多地依赖于各种复杂系统的运行，如因特网、移动通信网络、万维网、交通网络、电力网络、经济网络、社会关系网络等。这些系统可以抽象成一个复杂网络加以研究。不同于随机网络，复杂网络往往表现出很多特殊的属性，如节点度的幂律（power-law）分布、小世界（small world）、无标度（scale-free）、社团结构（community structure）等。分析网络的社团结构，不仅有助于在中观结构上（相对于在宏观结构上分析网络整体，或在微观结构上分析节点）理解

收稿日期：2017-11-01；**修回日期：**2017-12-28　**基金项目：**陕西省科技统筹创新工程计划资助项目（2016TZC-G-7-4）；陕西省自然科学基础研究计划资助项目（2016JM6048）；陕西省自然科学与技术研究计划资助项目（2016GY-092）；陕西省教育厅专项科学研究项目（16JK1687）

作者简介：孙韩林（1980-），男，陕西韩城人，副教授，博士，主要研究方向为复杂网络、大数据分析等（sunhanlin@xupt.edu.cn）；马素刚（1981-），男，高级工程师，硕士，主要研究方向为图像处理；王忠民（1967-），男，教授，博士，主要研究方向为大数据处理与应用、嵌入式系统、机器人技术等.

复杂网络的拓扑结构特性，还可用到具体应用中产生实际效益。例如，万维网络中，社团可以看做是一组相关主题的网页，社团分析结果可用于优化搜索结果；在线社会网络中，社团是一群具有密切联系的成员，社团分析结果有助于设计更加可靠的朋友推荐系统；购物网站中，通过对用户及其购买商品网络进行社团结构分析，可识别出具有相似购买兴趣的客户，从而建立更加精准的商品推荐系统等。

一个社团是网络中一组节点的集合，这些节点往往具有某些共同的属性或在网络中具有相似的功能；在拓扑结构上，表现为社团内部节点间具有更多的连接边，而社团成员节点与网络其余节点之间的连接边则相对较为稀少。给定一个网络，社团结构分析算法的目标就是找出这些节点集合。从现实世界抽象出来的复杂网络，其社团结构往往具有重叠（overlapping）特性，即一个节点可同时归属于多个社团。正确地识别出重叠结构是社团检测算法面临的一个挑战。

本文在以往工作[1]的基础上，提出了一种基于群体智能思想的自组织重叠社团结构分析算法 SO^2CSA^2（self-organizing overlapping community structure analysis algorithm）。群体智能最初由 Beni 和 Wang[2]共同提出，用于描述一组机器人的群体行为。群体智能可以看做是一种具有分布式、自组织特点的自然或人造系统的行为集合。在群体系统中，有大量具有相同行为和目标的简单个体，它们都是自治的，能与周围的邻居及系统环境进行交互。群体智能理论认为，由数量众多的简单个体作出的简单行为选择，将会导致整个群体表现出远远超过单个个体的智能，且这一现象是普遍存在的。SO^2CSA^2 算法把网络视为一个群体系统，网络节点是群体中的一个个个体；每个节点依据当前自己关于网络社团结构的局部知识自主地选择要加入哪些社团（可同时加入多个社团）；通过所有节点的迭代的加入行为选择，整个网络的社团结构将逐渐生长出来。

1 相关工作

社团结构分析算法依据的基本思想主要有凝聚、分裂、随机游走、信息扩散、谱分析、统计推断、优化算法等。文献[3,4]从不同角度对大多数算法进行了综述：文献[3]给出了多种社团元定义，以定义类别组织相关算法；文献[4]则按算法的原理进行了分析。文献[5,6]重点关注重叠社团结构的分析算法，尤其是文献[5]对比了 14 种典型算法的性能。文献[7]则综述了社会网络（social networks）的社团结构分析算法。本章简要介绍与本文最为相关的重叠社团结构分析算法，主要是通过社团演进从而获得网络社团结构的算法。

已有研究中，重叠社团结构分析可分为两种思路：一种是设计能直接从网络检测重叠社团的算法；另一种则是利用已有的非重叠社团检测算法，首先获得网络的一种非重叠社团结构，再通过调整社团边缘节点及其邻居的社团归属来最终获得一种重叠社团结构。

1.1 重叠社团直接检测算法

1）标签传播算法

标签传播算法（label propagation algorithm,LPA）的出发点是利用信息在具有社团结构的网络中的传播特性，即信息沿边以更高的概率在社团内部传播。首先给网络的每个节点分配一个标签（代表节点的归属社团）；此后，节点不断地将自己的标签传递给邻居，并从接收到的邻居标签中选择一个（如最多的共同标签）作为自己的新标签；当所有节点的标签不再改变时，算法结束，通过查看节点标签即可获得网络的一种社团结构：具有相同标签的节点归属于同一个社团。当允许节点同时拥有多个标签时，标签传播算法就可用于重叠社团结构分析。这类算法的最大优点是简单，具有近似线性的时间复杂度；其不足主要是准确度不高，且结果不稳定。用于重叠社团检测的标签传播算法主要有 COPRA[8]、BMLPA[9]、SLPA[10]、MLPA[11]、LPAcw[12]、DLPA[13]等。不同算法的区别主要在于标签选择策略以及传播策略不同。

2）局部扩展及优化算法

局部扩展及优化（local expansion and optimization）算法的基本思想可分为种子节点(集)扩展和局部社团合并两种类别。

种子节点（集）扩展方法中，从给定的一个或一组核心节点出发，逐步添加社团成员的邻居到社团中，只要邻居节点加入后能改善社团的质量，直到不存在这样的邻居节点为止，这样就检测到一个社团。然后在剩余节点中再次选择核心节点（集），检测新的社团。若选择新核心节点（集）后允许在整个网络上（而不仅仅是剩余网络上）检测其他成员节点，部分节点就可能会被同时包含到多个社团中，因而可用于重叠社团检测。这类算法发现的社团结构的质量依赖于种子节点（集）的选择。典型的种子节点（集）扩展算法主要有 IS2[14]、快速 LFM[15]、Moses[16]以及 OSLOM[17] 等。

局部社团合并方法中，首先在以节点为中心的局部网络上检测本地的局部社团（partial community），而后依据一定的策略（如局部社团的相似性、合并后社团的质量是否增加等）逐步合并局部社团，最终得到整个网络的社团结构。由于节点可能归属于多个局部社团，所以支持重叠社团的检测。某个节点的局部网络通常是以该节点为中心且包含其所有邻居节点及其连边的网络。算法的主要区别在于局部社团检测方法及合并策略不同。在局部网络上检测局部社团较为简单，算法的复杂度主要取决于局部社团的合并计算。这类算法主要有 DEMON[18]、PCMA[19]、EgoClustering[20]以及文献[21]的算法等。

3）基于博弈论的算法

基于博弈论的算法（game-theoretic algorithms）模拟了一种达到当前社团结构状态的演进过程。算法把网络中的每个节点看做是一个理性的（rational）或自私的（selfish）个体，每个个体依据定义的效用函数决定自己的社团归属。允许个体同时加入多个社团时，可用于重叠社团结构分析。根据博弈论理论，当效用函数是局部线性函数时，存在纳什均衡（Nash equilibrium）

状态，即没有个体可以通过单方面改变自己的行为来增加自己的效用函数。用于社团结构分析时，达到均衡状态表明得到了一种优化的社团结构。由于寻找全局纳什均衡是一个 NP-hard 问题，这类算法通常寻找一种局部纳什均衡作为替代。博弈论算法有严格的数学模型，但要求设计满足局部线性特性的效用函数（包括收益函数和损失函数），这可能会限制社团结构性质的表达。例如本文提出的连接分数就不满足局部线性特性，因而不能用做效用函数中的收益函数。这类算法主要包括 Game-Theoretic[22]、PSGAME[23]、NGGAME[23]、GADM[24]以及文献[25]的算法等。不同算法的主要区别在于效用函数的设计不同。

4）基于群体智能的算法

群体智能实际上包含有一系列算法。遗传算法（genetic algorithm，GA）和蚁群算法（ant colony algorithm, ACO）是两种用于社团结构分析的典型群体智能算法。遗传算法通过种群个体的交叉、变异以及选择等进化操作来求解问题的近似优化解。当应用于边网络（edge network）分析时，可发现网络的重叠社团结构[26]。但受限于进化个体的表示方法，基于 GA 的算法难以用于大规模网络的社团结构分析。文献[27]基于蚁群算法提出了一种重叠社团结构分析算法 AntCBO。实际上，该算法也是一种标签传播算法，只是在标签传播时采用了 ACO 算法的框架，即标签由在网络上爬行的蚂蚁携带到邻居节点。

在 EgoClustering 算法[20]的基础上，该文作者提出了一种重叠社团检测算法 COGS （community optimization Graph swarm）[28]。COGS 算法明确提及基于群体智能思想：将网络视为一个群体，从每个节点的视角出发，通过节点与其邻居的交互完成局部社团的发现，再结合标签传播思想实现局部社团的合并。COGS 中群体智能只是用于局部社团发现；与之不同，本文算法则是把群体智能作为一种模拟达到当前社团结构状态的框架。

实际上，标签传播类算法和博弈论类算法也可以看做是一种类型的群体智能算法：都是通过节点间的交互及迭代自主选择，从而演进出一种优化的网络社团结构。不同的是，标签传播类算法背后的依据是信息（标签）在社团内部传递的概率更高；而博弈论类算法则是寻求一种局部纳什均衡状态。此外，由于博弈论类算法和 SO^2CSA^2 算法都模拟了一种到达当前社团结构状态的过程，算法中节点都是直接选择要加入的社团，所以两者具有较大的相似性，其区别在于：a)博弈论类算法中，个体的策略（即要加入哪些社团）把所有要加入的社团作为一个整体考虑，用一个效用函数进行衡量，而 SO^2CSA^2 算法中，个体是否加入一个社团是分别加以考虑的；b) 博弈论类算法中，为了能有效进行计算，个体在受限的策略空间中进行选择，即个体的社团演进选择策略是：离开当前已加入的一个社团（leave），或加入一个尚未加入的邻居社团（join），或离开当前已加入的一个社团并加入一个尚未加入的邻居社团（switch），个体从中选择最优的一个执行，因而社团演进的速度较慢，而本文算法中个体分别考虑要加入的每一个邻居社团，因而策略选择更简单，社团的演进速度更快；c) 博弈论类算法具有严格的数学模型，要定义满足局部线性特性的收益函数和损失函数，已有算法中收益函数基于社团结构或节点相似度定义，而损失函数则简单定义为加入社团的“费用”（实际与加入的社团个数有关），但这种损失函数与社团结构没有直接关系，因此反而可能会影响社团的有效演进，而 SO^2CSA^2 算法是一种启发式算法，所依据的演进策略没有先置条件，更具一般性，因而能在演进策略中更加灵活地表达社团结构的特性。

1.2　基于非重叠社团检测的算法

这类算法的基本思想是：首先利用非重叠社团检测算法获得网络的一种非重叠社团结构；再以这些社团为基础，通过调整社团边缘节点及其邻居的社团归属来建立一种重叠的社团结构。文献[29]的算法通过比较社团边缘节点的离开或其邻居节点的加入前后社团质量是否有改善来发现网络的重叠社团结构。文献[30]分析了若干真实网络的社团结构，发现用非重叠检测算法发现的社团结构在去除重叠节点后，与网络真实社团结构去除重叠节点后的结构具有很强的相似性，进而提出 POVC （permanence based vertex-replication algorithm for overlapping community detection）算法。在非重叠社团结构的基础上，该算法依据节点复制持久性（permanence based vertex-replication）指标[31]，尝试把社团边缘节点加入其邻居所属的社团，从而发现网络的重叠社团结构。

这类算法的性能在很大程度上依赖于非重叠检测算法能否在具有重叠社团结构的网络上有效地检测到社团的边界。此外，仅顺序调整一次边缘节点及其邻居的社团归属的方式也忽略了节点社团归属之间可能存在的关联性。

2　算法描述

2.1　社团连接分数

算法 SO^2CSA^2 基于群体智能思想模拟一种到达当前社团结构状态的演化过程。社团演化过程中，节点是否加入一个社团主要依据节点与该社团的连接分数。

文献[31]在分析多个真实网络的社团结构的基础上，提出节点是否归属于一个社团取决于两种因素：a)节点与每个外部社团（即未加入社团）的连接的数量，而不是与所有外部社团的连接的总数量；b)节点与所属社团的连接强度，而不仅仅是与所属社团内部节点的连接的数量（即属于同一社团的邻居的数量），连接强度则用属于该社团的邻居节点的聚类系数表示，聚类系数越大，连接强度越大，反之越小。该文提出了节点复制持久性指标作为节点加入社团的依据，其定义如式(1)所示。

$$Perm(v)=\left[\frac{I(v)}{E_{\max}(v)}\times\frac{1}{D(v)}\right]-\left[1-C_{in}(v)\right], \tag{1}$$

其中： $I(v)$ 表示节点与所属社团的内部连接数； $E_{\max}(v)$ 表示节点与外部社团的最大外部连接数； $D(v)$ 是节点度； $C_{in}(v)$ 是同属于节点所属社团的邻居节点间的聚类系数。式中的第 1 项度量了第一种因素；而第 2 项实际上度量了第二种因素，即根

据聚类系数对第一种因素施加一定的惩罚，聚类系数越大，惩罚越小。节点复制持久性的取值范围是（-1,1]，值越大，表明节点对该社团的归属度越高。

本文提出一种新的节点的社团归属度量指标连接分数（connection score, CS），其定义如式（2）所示。

$$CS(v)=\left(\frac{I(v)}{D(v)}\right)^{[1-C_{in}(v)]}, \tag{2}$$

其中：$I(v)$、$D(v)$ 和 $C_{in}(v)$ 与节点复制持久性定义中相同（式中也可用 $E_{\max}(v)$ 代替 $D(v)$）。连接分数认为，节点是否属于一个社团，首先取决于节点与该社团内部连接的数量在节点总连接数（度）中所占的比例（或内部连接数量与最大外部连接数量的比例）；其次取决于同属于该社团的节点邻居之间的聚类系数；聚类系数对连接分数具有强化作用，而不是作为一种可叠加的因素单独考虑。连接分数的取值范围是[0,1]，值越大，表明节点对该社团的归属度越高。

2.2 基于群体智能的重叠社团结构分析算法

SO^2CSA^2 算法分为三个步骤，即社团结构的初始化、社团结构的演进以及社团结构的调整。

2.2.1 社团结构的初始化

算法从一种初始的社团结构开始，迭代地进行各节点社团归属的演进。初始社团结构的质量对算法运行效率有较大影响：在一种高质量的初始社团结构的基础上，网络的社团结构能够较快地演进到一种优化的结果。从社团的核心特征（社团内部联系紧密，而外部连接相对稀疏）出发，初始化算法在网络中寻找一组 K-派系（K-clique）作为初始社团。K-派系的成员节点间具有最紧密的联系，即完全连接，因而可以确信它们属于同一个社团。

初始化社团结构的生成过程如算法 1（Initialize）所示。算法简单地为每个节点只寻找一个初始社团。当节点有邻居已经被初始化，且节点加入后该社团仍是一个 K-派系时，节点加入该社团；否则，如果节点能与两个尚未初始化的邻居形成一个 3-派系，这三个节点就组成一个新的初始社团。如果上述两种条件都不能满足，节点就形成一个单独社团。初始化算法生成的初始社团要么是一个 K-派系，要么是一个单独节点，其中 K-派系是下一步社团结构演进的核心。

```
算法 1  Initialize
输入：网络。
输出：初始社团结构。
过程：
while (有未初始化的节点 n）{
  if（n 有邻居已初始化 && n 与该邻居所属社团的所有成员构成 K-派系）{
    n 加入该社团；
    标记 n 为已初始化；
  } elseif（n 与未初始化的两个邻居构成 3-派系）{
    n 与该两邻居形成一个新社团；
    标记该 3 个节点为已初始化；
  } else {
    n 单独形成一个社团；
    标记 n 为已初始化；
  }
}
```

2.2.2 社团结构的演进

在初始社团结构的基础上，所有节点不断地演进自己的社团归属，网络的社团结构将会逐渐生长出来。节点社团演进的基本思想是：a) 节点只能加入有其邻居存在的社团（称为候选社团）；b) 节点通过与邻居的交互获得这些可能加入的候选社团，并自主决定加入哪些社团。

节点主要依据与候选社团的连接分数来决定是否加入该社团。由于计算时要用到聚类系数，只有在节点与候选社团的连接数大于或等于 3 时才能计算连接分数；当连接数小于 3 时，是否加入需要作特殊处理。节点获得所有候选社团后，是否加入某个社团的规则如下：

a) 若与该社团的连接分数最大，则加入该社团；

b) 若连接分数与最大连接分数的比率（简称连接分数比）超过给定的阈值，加入该社团；

c) 若节点与该社团的连接数为 2(此时不能计算连接分数)，且与所有候选社团的最大连接数小于或等于 3，加入该社团；

d) 若不存在满足上述条件的候选社团，节点就单独形成一个社团。

如果同时有多个候选社团满足上述加入条件，算法还将检查这些社团中是否有某些（小）社团被包含在另一个（大）社团中。如果存在这种情况，大社团将吸收小社团（即删除小社团），以加速算法的收敛。

当没有节点的社团归属发生改变时，社团结构的演进结束。但通常这种理想状态难以达到，算法通过设置最大演进次数来确保运行终止。

设团结构的演进如算法 2（Evolve）所示。

```
算法 2  Evolve
输入：网络；初始社团结构；连接分数比阈值 t。
输出：演进的社团结构。
过程：
while（当前演进次数 ≤ 最大演进次数）{
  生成一个随机的节点演进顺序；
  for（按演进顺序的每一个节点 n）{
    获取 n 的候选社团；
    计算 n 与每个候选社团的连接分数；
    for（每一个候选社团 cc）{
      if（n 与 cc 的连接数 > 2）
        if（n 与 cc 的连接分数最大）
          n 加入社团 cc；
        elseif（n 与 cc 的连接分数比 ≥ t）
```

```
        n 加入社团 cc；
      elseif((n 与 cc 的连接数==2）&&
        n 与所有候选社团的最大连接数 ≤ 3))
        n 加入社团 cc；
   }
   if（n 未加入任何候选社团）
     n 单独形成一个社团；
   if（n 加入的社团有改变）
     n 通知其邻居节点；
   if（n 加入了多个社团）
     while（有社团包含在另一个社团中）{
       删除被包含的社团；
       通知被删除社团成员及其邻居；
     }
 }
 if（没有节点的社团归属发生改变）
   return;
}
```

群体智能算法中，每个个体都保存有自己的状态，需要保证这些状态的一致性。这通过个体间的交互实现。对 SO^2CSA^2 算法，每个节点都保存了自己加入的社团，因而当社团成员发生变化时，必须保持所有成员节点存储的社团副本一致。为了减少个体交互开销，算法实现时，采用共享方式存储社团结构，而每个节点只保存其加入社团的引用标志；这样，当社团归属发生改变时，节点就不需要通知要加入或离开社团的其他成员节点：它们都引用了同一个社团副本，而社团引用标志不发生变。

此外，为了简化个体交互，SO^2CSA^2 算法采用了主动通知的方式，即当社团归属发生变化时，节点主动通知其邻居自己新加入哪些社团以及离开哪些社团。邻居节点可根据这些通知来计算自己的候选加入社团，而不再需要额外的交互。特别需要注意的是，在删除被包含的小社团时，节点需要通知被删除社团的所有成员及其邻居节点，以保证这些节点状态的一致性。

2.2.3 社团结构的调整

社团结构演进结束后，某些节点可能会加入错误的社团，或未加入应加入的社团。例如，由于受演进次数限制，以及节点各自独立地更新其归属社团等原因，节点可能与已加入的某些社团之间的连接数很小（甚至为 0），但尚未及时更新；虽然节点与社团的连接数很大（甚至超过当前加入社团的最大连接数），但由于聚类系数小，连接分数比达不到阈值，所以节点未加入这些社团。因此，在社团结构演进完成后，需要对获得的社团结构进行后处理，即调整部分节点的社团归属，以提高社团结构的质量。

后处理可根据社团结构的具体特点进行。实验部分针对合成网络和真实网络的社团结构分别设计了不同的后处理算法。

3　社团结构质量的度量

3.1　归一化互信息

当已知网络的真实社团结构时，可用归一化互信息（normalized mutual information, NMI）来度量算法检测到的社团结构与真实社团结构的相似程度，从而评判检测社团结构的整体质量。归一化互信息值越大，表示算法发现的社团结构质量与真实社团结构越相似，社团结构质量越好。其取值范围是[0,1]，取 1 时表示算法发现的社团结构与真实社团结构完全相同，而取 0 时则表示两者完全不同。文献[32]提出了一种用于重叠社团结构相似性计算的归一化互信息计算方法，其定义如式（3）所示。

$$NMI = \frac{I(X:Y)}{\max(H(X),H(Y))}, \tag{3}$$

其中：X 和 Y 分别表示两种社团结构；$H(X)$ 和 $H(Y)$ 是两种社团结构的熵；$I(X:Y)$ 是 X 和 Y 的互信息，定义为

$$I(X:Y) = \frac{1}{2}\left[H(X) - H(X|Y) + H(Y) - H(Y|X)\right], \tag{4}$$

其中：$H(X|Y)$ 和 $H(Y|X)$ 是 X 和 Y 的一种条件熵（更详细信息参见文献[32]）。

3.2　模块度密度

现实世界网络的真实社团结构多是由人手工建立的，实际上往往难以获得，尤其是对大规模网络而言。在不知道真实社团结构的情况下，Newman 提出的模块度（modularity）被广泛用于（非重叠）社团结构质量的度量。有多个研究将 Newman 模块度进行了扩展，用于度量重叠社团结构的质量；文献[33]对这些扩展进行了对比分析。

Newman 模块度存在偏好（bias）问题：在一些情况下偏好小社团，而在另一些情况下偏好大社团；其中后者在文献中称为模块度的分辨率问题[33]。因此 Chen 等人提出了模块度密度（modularity density）的概念，用于解决偏好问题。文献[33]也将模块度密度进行了扩展，提出了一种适用于重叠社团结构的模块度密度计算方法，其定义如式（5）所示。

$$Q_{ds}^{ov} = \sum_{c \in C}\left[\frac{|E_c^{in}|}{|E|}d_c - \left(\frac{2|E_c^{in}| + |E_c^{out}|}{2|E|}d_c\right)^2 - \sum_{c' \in C, c' \neq c}\frac{|E_{c,c'}|}{2|E|}d_{c,c'}\right] \tag{5}$$

其中：C 是一种社团结构；c 和 c' 是其中的社团；其余变量的定义如下：

$$|E_c^{in}| = \frac{1}{2}\sum_{i,j \in c} f(a_{i,c}, a_{j,c}) A_{ij},$$

$$|E_c^{out}| = \sum_{i \in c}\sum_{c' \in C, c \neq c', j \in c'} A_{ij},$$

$$|E| = \frac{1}{2}\sum_{ij} A_{ij},$$

$$|E_{c,c'}| = \sum_{i \in c, j \in c'} f(a_{i,c}, a_{j,c'}) A_{ij},$$

$$d_c = \frac{2\left|E_c^{in}\right|}{\sum_{i,j\in c,i\neq j} f\left(a_{i,c},a_{j,c'}\right)},$$

$$d_{c,c'} = \frac{\left|E_{c,c'}\right|}{\sum_{i\in c,j\in c'} f\left(a_{i,c},a_{j,c'}\right)},$$

其中：A_{ij} 是网络邻接矩阵的元素，若节点 i 和 j 相连接，则为1，否则为0；$a_{i,c}$ 是节点 i 归属于社团 c 的归属系数（belonging coefficient）；$f\left(a_{i,c},a_{j,c}\right)$ 是一种归属函数（belonging function），如可以是 $a_{i,c}$ 和 $a_{j,c}$ 的乘积或平均值等。从有效性和简单性出发，文献[33]推荐 $a_{i,c}$ 取为节点重叠度的倒数，而归属函数则为两者的乘积。

3.3　重叠节点的质量

归一化互信息和模块度密度都是从统计意义上度量了社团结构的整体质量，但无法给出重叠节点质量的确切指示。重叠节点的质量包含两个层面：a)是否正确地发现了重叠节点，即发现的重叠节点确实是重叠节点，以及真正的重叠节点都能够被发现；b)发现的重叠节点的重叠度（即节点加入的社团个数）是否正确。在已知重叠节点的情况下，可以用召回率（recall，R）、准确率（precision， P）以及两者的调和平均值 F-分数（F-score， F）来指示是否正确地发现了重叠节点，其定义分别如式（6）～（8）所示。

$$R = \frac{\text{算法正确发现的重叠节点的数量}}{\text{真实重叠节点的数量}}, \tag{6}$$

$$P = \frac{\text{算法正确发现的重叠节点的数量}}{\text{算法发现的重叠节点的总数量}}, \tag{7}$$

$$F = \frac{2\times R\times P}{R+P}。 \tag{8}$$

重叠节点重叠度的质量可用重叠度的分布来指示。

4　实验及结果分析

本文用若干合成网络和现实世界网络来验证 SO^2CSA^2 算法检测重叠社团结构的性能。文献[5]的大量对比实验表明，检测重叠社团结构时，SLPA 和 OSLOM 算法具有较好的性能，且这两种算法的源代码可公开获得（SLPA，https://sites.google.com/site/communitydetectionslpa/；OSLOM，http://www.oslom.org/）。本文也对比了 SO^2CSA^2 与这两种算法的实验结果。

4.1　网络数据

4.1.1 合成网络

验证社团结构分析算法的性能时，最基本的方法是对比合成网络的真实社团结构与算法发现的社团结构，计算两者的相似程度。LFR 模型[34]是最常用的合成网络模型。实验中，生成节点数（参数 N）为 10 000 的网络，其节点平均度（参数 k）为 40，节点最大度（参数 kmax）100，社团最大规模（参数 cmax）和最小规模（参数 cmin）分别为 100 和 20，节点度分布负指数（参数 t1）为-2，社团规模分布负指数（参数 t2）为 -1。在上述参数设置下，社团混合参数（μ）分别取 0.1、0.2、0.3、0.4 及 0.5，节点重叠度（参数 om）取 2、4、6、8 及 10，重叠节点数（参数 on）取 1 000、2 000、3 000、4 000、5 000、6 000、7 000 及 8000，共生成 18 个合成网络。混合参数 μ 指明了社团成员与外部网络的连接数占该社团成员总连接数的比率，值越大，表明社团结构越模糊。

4.1.2 现实世界网络

合成网络的拓扑性质可通过参数进行控制，但现实世界网络的拓扑性质往往比合成网络更复杂。实验中也用三 个从现实系统抽象的网络来验证算法的性能，分别是 Email[35]、Power[36] 和 PGP[37]网络。其中 Email 网络是西班牙罗维拉—威尔吉利大学（Univeristy Rovira i Virgili）的电子邮件交互网络； Power 网络是美国西部的电力网络； PGP 网络是使用 PGP 算法（pretty-good-privacy algorithm）进行安全信息交换的部分用户构成的一个连通网络。这三个网络均是无向、无权重的网络。

若网络中一个节点的度为 1，即该节点只与唯一一个节点连接，则如果该节点要加入社团，它只能加入到所连接节点所在的社团。实验中首先对这些网络进行预处理，即迭代地移除度为 1 的节点，直到不存在这样的节点为止；再应用算法对其进行社团结构分析。移除这类极端状态节点，可减小网络的规模，加快算法的运行。经过预处理后，Email、Power 和 PGP 网络分别移除了 13.68%、32.14%和 49.12%的节点，但却保留了97.16%、75.92%和 78.43%的边，即经过预处理修正后的网络保留了原网络的核心拓扑结构特征。这些现实世界网络及其修正网络的属性如表 1 所示。

表 1　现实世界网络的属性

网络	节点数	边数	修正后节点数	修正后边数
Email	1133	5451	978	5296
Power	4941	6594	3353	5006
PGP	10680	24316	5434	19070

4.2　实验设置

实验中，对每一个网络三种算法各运行 30 次，最后分别计算 30 次运行结果的统计值（平均值及标准差）。

运行 SO^2CSA^2 时，最大演进次数设置为 20；连接分数比阈值的设置采用渐进策略，即在最初若干次演进中，该阈值取较大值，以获得更准确的核心社团，而在随后的演进中降低该阈值，从而得到扩展的、更完整的社团。实验中通过多次尝试设置了合理的连接分数比阈值：对合成网络，前 10 次演进该阈值设为 0.8，后 10 次设为 0.65；对真实网络，该阈值始终设置为 0.9。

运行 SLPA 算法时，其参数 r 遍取所有默认值，并从中选择最好的（归一化互信息值或模块度密度值最大的）结果。OSLOM 算法参数取默认值。

4.3 合成网络的实验结果分析

4.3.1 后处理

合成网络社团结构的后处理简单地依据节点与社团的连接数进行社团归属调整。实际上，这种后处理可以看做是一种归属条件更为宽松的社团结构演进。后处理过程如算法 3（Postprocess-1）所示。对已经加入的社团，如果节点与该社团连接数小于其与所有有连接社团的最大连接数（简称为最大社团连接数）的 1/2，节点离开该社团；对没有加入的候选社团，如果节点与该社团连接数大于或等于最大社团连接数的 1/2，则节点加入该社团；当没有节点的社团归属发生改变时，算法结束。类似地，采用设置最大调整次数的方法确保算法终止。实验中最大调整次数设置为 3。

```
算法 3　Postprocess-1
输入：网络；社团结构。
输出：修正的社团结构。
过程：
while（调整次数 ≤ 最大调整次数）{
  生成一个随机的节点处理顺序；
  for（按处理顺序的每个节点 n）{
    计算 n 与所有有连接社团的连接数；
    for（每一个有连接社团 cc）
      if(n 已加入 cc && 与 cc 的连接数小于最大社团连接数的 1/2)
        n 离开 cc；
      elseif（n 未加入 cc && 与 cc 的连接数大于最大社团连接数的 1/2）
        n 加入 cc；
  }
  if（没有节点更新社团归属）
    return；
}
```

4.3.2 社团结构的整体质量

图 1 给出了三种算法检测到的重叠社团结构的平均归一化互信息及标准差。从图 1 中可以看到：a) 当混合参数 μ 从 0.1 增加到 0.5 时（取 om=3，on=1000），算法 SO^2CSA^2 的平均归一化互信息值超过了 SLPA 和 OSLOM 两种算法，接近 1，表明 SO^2CSA^2 发现的社团结构与真实社团结构更相似，即社团结构质量更好；b)当节点重叠度 om 从 2 增加到 10 时（取 μ=0.3，on=1000），三种算法的平均归一化互信息值均逐渐减小，但 SO^2CSA^2 的减小速度最慢，远好于其他两种算法，表明 SO^2CSA^2 在检测节点重叠度方面更具优势；c) 当网络重叠节点的数量 on 从 1 000 增加到 8 000 时（取 μ=0.3，om=3），总体上还是 SO^2CSA^2 发现的社团结构质量更好；SLPA 和 SO^2CSA^2 的平均归一化互信息值均逐渐减小，但 SO^2CSA^2 的减小速度更慢；而 OSLOM 的平均归一化互信息值先是减小，但当 on 取值超过 6 000 时反而有缓慢的上升，甚至在 on 取 8000 时超过了 SO^2CSA^2，表明 OSLOM 算法更适合于分析深度重叠网络，即有大比例重叠节点的网络；d)在 μ 和 om 变化的两种情形下，SLPA 发现的社团结构质量比 OSLOM 要好；在 on 变化的情形下，OSLOM 的平均归一化互信息值先减小后缓慢增加，这些现象都与文献[5]中的实验结果相一致。

SO^2CSA^2 算法在重叠节点比例很高时性能变差，原因在于这种情形下可能有部分社团之间的差异较小，如果给定的初始社团是这些相似社团的公共部分，则其往往只能演进为一个社团。本文对比了 on 取 8000 时算法 OSLOM 和 SO^2CSA^2 检测到的社团结构，发现在所有实验结果中，后者检测到的社团数量总是较少，这在一定程度表明上述原因分析是合理的。

由于无法获得基于博弈论的社团检测算法的源代码，同时为了避免算法实现带来的误差，本文对比了 SO^2CSA^2 算法的结果与文献[5,22,25]中博弈论类算法分析 LFR 合成网络社团的结果。本文实验中设置的 LFR 参数（包括节点数、节点度分布、社团大小分布、重叠节点比例及节点重叠度等）更为严苛；对比相近参数设置（对 SO^2CSA^2 重叠度是 3，平均节点度 40，最大节点度 100；而对博弈论类算法重叠度是 2，平均节点度 20，最大节点度 50，其余参数相同；后一设置下社团发现更容易）情况下的归一化互信息，发现 SO^2CSA^2 算法的 NMI 值更大，社团结构质量更好。

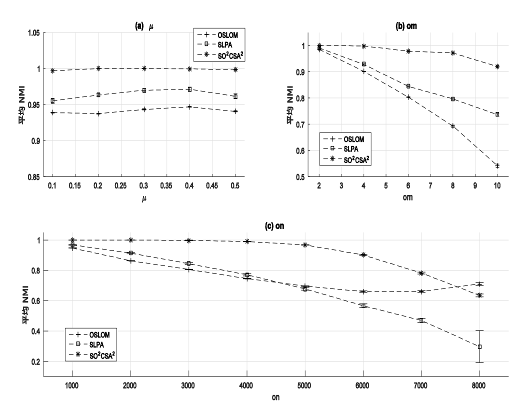


图 1　合成网络的平均归一化互信息及其标准差

4.3.3 重叠节点的质量

图 2 给出了三种算法发现的社团结构中重叠节点的平均召回率及标准差。从图 2 中可以看出：a) 在各种情形下，SO^2CSA^2 算法的召回率几乎都是最好的，表明该算法能有效地检测到重叠节点；b) 多数情况下（除网络重叠节点比例很高时），SLPA 算法的重叠节点召回率次好；c)大多数情况下，OSLOM 算法的重叠节点召回率最差，尤其是当节点重叠度较大时，但在用于深度重叠网络（网络重叠节点比例很高）分析时有所改善，其重叠节点召回率逐渐上升。

图 3 给出了三种算法发现的社团结构中重叠节点的平均准确率及标准差。从图 3 中可以看出：a) μ 值增大的过程中，三种算法的重叠节点准确率相差不大，当 μ 值较大时，SO^2CSA^2 算法的准确率更高；b) 节点重叠度 om 从 2 增大到 10 时，SLPA 的准确率最好，其余两种算法的准确率较差，而在重叠度较大时，SO^2CSA^2 的准确率比 OSLOM 要好；c)重叠节点数 on 从 1

000 增加到 8 000 的过程中，算法 OSLOM 的准确率最好，SO^2CSA^2 次之（很接近），而 SLPA 的准确率在重叠节点比例增大时逐渐变差。

图 4 给出了综合召回率和准确率的平均 F-分数及标准差。从图 4 中可以看出：a) 总体来看，在 μ 或 on 变化时，SO^2CSA^2 算法发现的重叠节点的质量（节点是否重叠）最好，SLPA 算法次之（除分析深度重叠网络时外），OSLOM 算法最差（分析深度重叠网络时有所改善），注意 on 取 8 000 时没有 SLPA 的 F-分数值，这是因为在 30 次算法运行中，有一次运行结果的召回率和准确率都为 0，因而无法计算 F-分数；b)om 值变化时，从 F-分数值看，SLPA 发现的重叠节点的质量（节点是否重叠）最好，但这并不意味着它发现的社团结构的质量是最好的（从平均归一化互信息值看，SO^2CSA^2 要远好于 SLPA），实际上，尽管 SLPA 能更准确地发现重叠节点，但其发现的重叠节点度与真实值相比偏小，导致其社团结构质量变差。

图 5~7 分别给出了 om 取不同值时三种算法检测到的重叠节点重叠度的分布（算法 30 次运行的频数统计）。从图中可以看出，随着 om 值增大，三种算法发现的节点重叠度与真实节点重叠度的偏离越来越大（LFR 模型生成的网络中，所有重叠节点的重叠度都相同），但 SO^2CSA^2 的偏离程度最小，所能发现的正确节点重叠度的比例远高于其他两种算法，SLPA 和 OSLOM 发现的节点重叠度总体上比真实重叠度偏小。

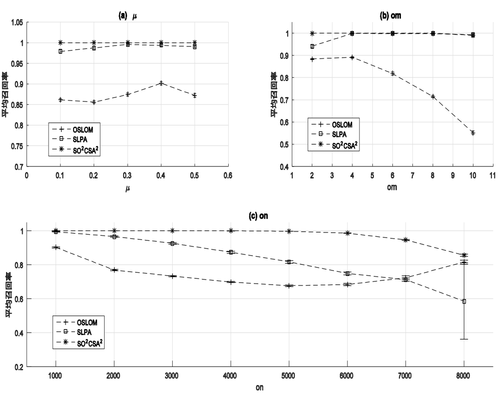


图 2　合成网络的平均重叠节点召回率及其标准差

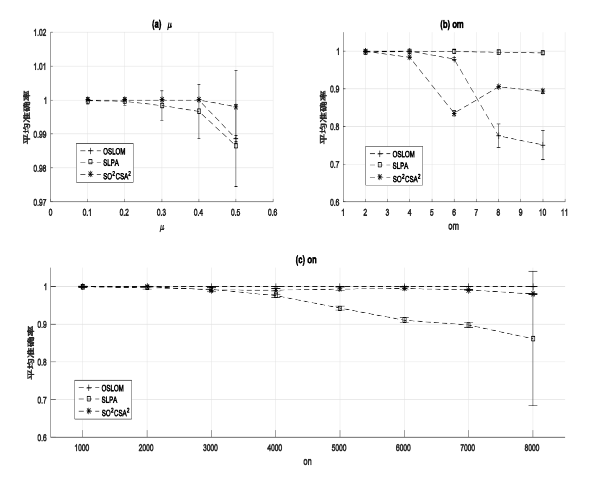


图 3　合成网络的平均重叠节点准确率及其标准差

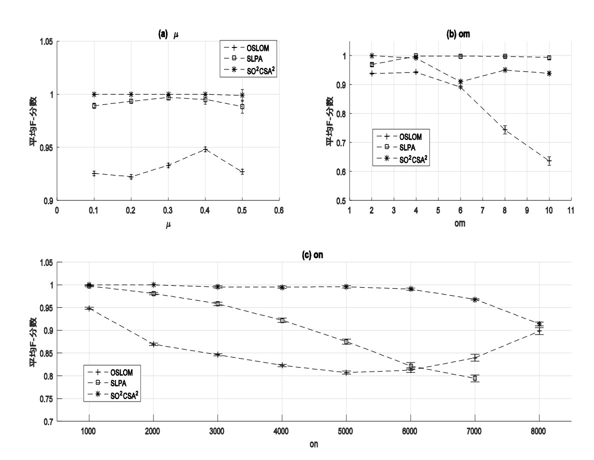


图 4　合成网络的平均重叠节点 F-分数及其标准差

上述合成网络的实验结果表明，无论从检测重叠节点还是从正确发现重叠节点的重叠度方面，SO^2CSA^2 算法都要优于两种对比算法。

4.4　真实网络的实验结果分析

4.4.1 后处理

通过分析 SO^2CSA^2 算法发现的现实世界网络的社团结构，发现其中包含大量只有一个成员节点的社团，且该节点与其他社团的连接数都为 1。从式（4）可以看出，这类节点对模块度密度的贡献是负值。如果社团结构中这类节点的数量较多，它们对计算的模块度密度可能产生很大影响，导致模块度密度值难以客观地反映更重要的、有多个成员的社团的质量。因此，在对比不同算法的性能时，如果不同算法发现的社团结构中这类节点的数量差异很大，则不能用直接计算的模块度密度进行比较。由于 SLPA 和 OSLOM 算法发现的社团结构中只有一个成员的社团极少，所以在比较之前需要对获得的社团结构进行修正。

本文对 SO^2CSA^2 算法发现的社团结构首先进行算法 3 的后处理，调整错误的节点社团归属，然后再进行算法 4（Postprocess-2）的修正，即移除这类只有一个成员节点、且该节点与其他社团的连接数为 1 的社团。称经过算法 4 处理后得到的社团结构为网络的核心社团结构（core community structure）。

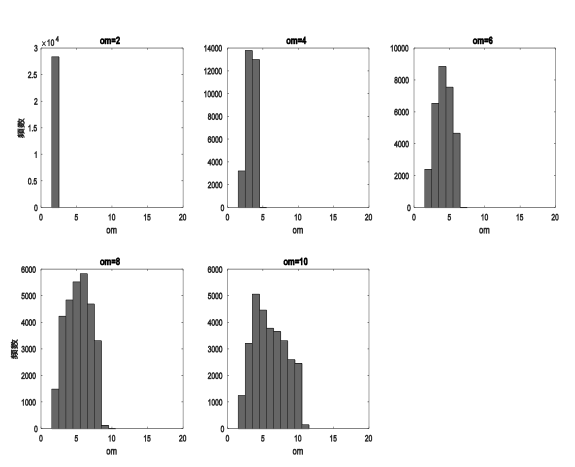


图 5　合成网络的重叠节点重叠度分布—SLPA 算法

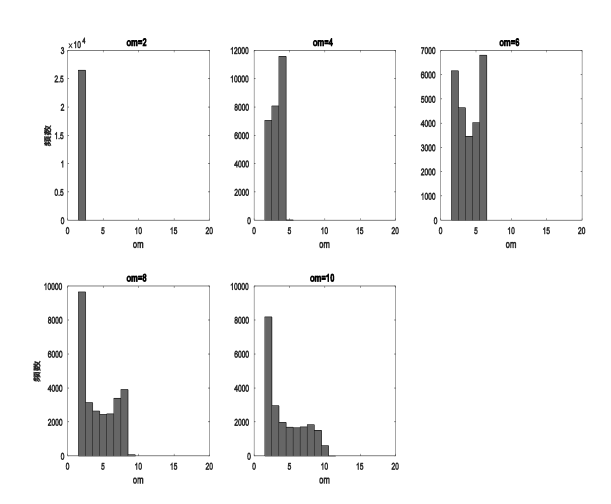


图 6　合成网络的重叠节点重叠度分布—OSLOM 算法

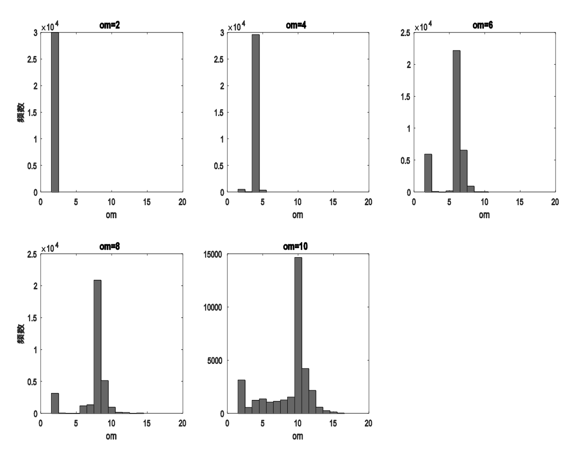


图 7　合成网络的重叠节点重叠度分布—SO²CSA² 算法

算法 4　Postprocess-2

输入：网络；社团结构。

输出：核心社团结构。

过程：

```
while（有只有 1 个成员的社团 c）{
  if（c 的唯一成员与其他社团的连接数都是 1）
    从社团结构中移除该社团;
}
```

对比不同算法的性能时，只能基于相同的网络社团结构进行。因此，本文也从 SLPA 和 OSLOM 发现的社团结构中移除了在 SO²CSA² 后处理中移除的节点。由于 SO²CSA² 30 次运行的结果略有差异，本文只移除多次运行结果中的公共移除节点，这部分节点占移除节点总数的 90%以上。在移除这些节点后，SLPA 和 OSLOM 算法发现的社团结构的模块度密度都增大了，表明上述处理是合理的。

4.4.2 核心社团结构的整体质量

图 8 给出了三个现实世界网络核心社团结构的平均模块度密度及标准差。从图 8 中可以看出：a)SO²CSA² 发现的核心社团结构具有更好的模块度密度，尤其是 Power 网络；b) SO²CSA² 的标准差最小，表明算法的稳定性更好。

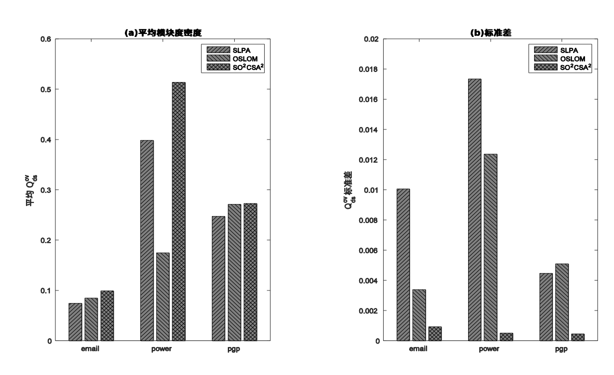


图 8　现实世界网络核心社团结构的平均模块度密度及其标准差

4.4.3 核心社团结构的重叠节点重叠度分布

图 9 给出了三种算法发现的现实世界网络核心社团结构的主要重叠节点重叠度分布。从图 9 中可以看到，对三个网络，SLPA 仅能发现重叠度为 2 的重叠节点；而 SO²CSA² 和 OSLOM 能够发现重叠度更大的重叠节点，尤其是 SO²CSA²，大多数情况下能发现更多的大重叠度重叠节点。基于合成网络实验中 SO²CSA² 能发现更准确的节点重叠度，本文认为 SO²CSA² 发现的现实世界网络的节点重叠度质量也更好。

综上所述，SO²CSA² 是一种有效的重叠社团结构分析算法。大多数情况下，所发现的社团结构无论从整体质量还是从重叠节点质量看，都比 SLPA 和 OSLOM 算法有较大的提升，尤其用于分析重叠节点比例较高或节点重叠度较大的网络时，性能提升更为显著。

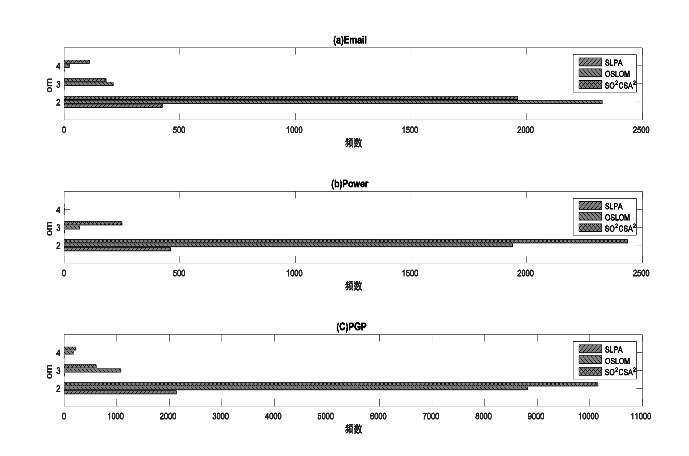


图 9　现实世界网络核心社团结构的重叠节点重叠度分布

5　结束语

本文基于群体智能思想提出了一种自组织的重叠社团结构分析算法 SO²CSA²。该算法模拟了网络社团结构到达当前状态的一种过程，分为三个阶段，即社团结构初始化、社团结构演化以及社团结构调整阶段。在初始化阶段，从社团内部联系紧密而外部连接相对稀疏的特性出发，算法在网络中寻找一组 K-派系作为初始的社团结构。在社团结构演化阶段，在初始社团结构的基础上，所有节点迭代地演进自己的社团归属，最终整个网络的社团结构逐渐生长出来；节点主要依据与社团的连接

分数来决定自己的社团归属。在社团结构调整阶段，通过后处理操作调整少量节点的社团归属，以改善社团结构的质量。在多个合成网络和现实世界网络上进行的一系列实验表明，SO^2CSA^2 算法能更有效地发现网络的重叠社团结构，尤其适用于分析重叠节点比例较高和重叠节点重叠度较大的网络。

参考文献：

[1] Sun Hanlin, Jie Wei, Sauer C, *et al.* A self-organizing algorithm for community structure analysis in complex networks [C]// Proc of the 17th IEEE//ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel//Distributed Computing. Washington DC: IEEE Computer Society, 2016: 5-12.

[2] Beni G, Wang Jing. Swarm intelligence in cellular robotic systems [C]// Proc of NATO Advanced Workshop on Robots and Biological Systems. Berlin: Springer, 1989: 703-712.

[3] Coscia M, Giannotti F, Pedreschi D. A classification for community discovery methods in complex networks [J]. Statistical Analysis and Data Mining, 2011, 4 (5): 512.

[4] Fortunato S, Community detection in graphs [J]. Physics Reports. 2010, 486 (3~5): 75.

[5] Xie Jierui, Kelley S, Szymanski B K. Overlapping community detection in networks: the state-of-the-art and comparative study [J]. ACM Computing Surveys, 2013, 45 (4): 43.

[6] Amelio A, Pizzuti C. Overlapping community discovery methods: a survey [M]// Social Networks: Analysis and Case Studies. Vienna: Springer, 2014: 105-125.

[7] Plantié M, Crampes M. Survey on social community detection [M]// Ramzan N, Zwol R v, Lee J S, Clüver K, Hua Xiansheng. Social Media Retrieval. London: Springer, 2013: 65-85.

[8] Gregory S. Finding overlapping communities in networks by label propagation [J]. New Journal of Physics, 2010, 12 (10): 103018.

[9] Wu Zhihao, Lin Youfang, Gregory S, *et al.* Balanced multi-label propagation for overlapping community detection in social networks [J]. Journal of Computer Science and Technology, 2012, 27 (3): 468.

[10] Xie Jierui, Szymanski B K, Liu Xiaoming. SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process [C]// Proc of the 11th IEEE International Conference on Data Mining Workshops. Washington DC: IEEE Computer Society, 2011: 344-349.

[11] Dai Qiguo, Guo Maozu, Liu Yang, *et al.* MLPA: detecting overlapping communities by multi-label propagation approach [C]// Proc of IEEE Congress on Evolutionary Computation. Washington DC: IEEE Computer Society, 2013: 681-688.

[12] Liang Zongwen, Li Jianping, Yang Fan, *et al.* Detecting community structure using label propagation with consensus weight in complex network [J]. Chinese Physics B, 2014, 23 (9): 098902.

[13] Sun Helia, Huang Jianbin, Tian Yongqiang, *et al.* Detecting overlapping communities in networks via dominant label propagation [J]. Chinese Physics B, 2015, 24 (1): 018703.

[14] Baumes J, Goldberg M, Magdon-Ismail M. Efficient identification of overlapping communities [C]// Proc of IEEE International Conference on Intelligence and Security Informatics. Washington DC: IEEE Computer Society, 2005: 27-36.

[15] Li Yanan, Zhu Zhengyu. A fast method of detecting overlapping community in network based on LFM [J]. Journal of Software, 2015, 10 (7): 825.

[16] McDaid A, Hurley N. Detecting highly overlapping communities with model-based overlapping seed expansion [C]// Proc of International Conference on Advances in Social Networks Analysis and Mining. Washington DC: IEEE Computer Society, 2010: 112-119.

[17] Lancichinetti A, Radicchi F, Ramasco J J, *et al.* Finding statistically significant communities in networks [J]. PLoS ONE, 2011, 6 (4): 0018961.

[18] Coscia M, Rossetti G, Giannotti F, *et al.* Uncovering hierarchical and overlapping communities with a local-first approach [J]. ACM Trans on Knowledge Discovery from Data, 2014, 9 (1): 6.

[19] Xu E H W, Hui P M. Efficient detection of communities with significant overlaps in networks: partial community merger algorithm [EB/OL]. (2017-09-28) [2017-10-26]. https://arxiv. org/abs/1509. 00556.

[20] Rees B S, Gallagher K B. Overlapping community detection by collective friendship group inference [C]// Proc of International Conference on Advances in Social Networks Analysis and Mining. Washington DC: IEEE Computer Society, 2010: 375-379.

[21] Shang Ronghua, Luo Shuang, Li Yangyang, *et al.* Large-scale community detection based on node membership grade and sub-communities integration [J]. Physica A: Statistical Mechanics and its Applications, 2015, 428: 279.

[22] Chen Wei, Liu Zhenming, Sun Xiaorui, *et al.* A game-theoretic framework to identify overlapping communities in social networks [J]. Data Mining and Knowledge Discovery, 2010, 21 (2): 224.

[23] Alvari H, Hashemi S, Hamzeh A. Detecting overlapping communities in social networks by game theory and structural equivalence concept [C]// Proc of International Conference on Artificial Intelligence and Computational Intelligence. Berlin: Springer, 2011: 620-630.

[24] Hajibagheri A, Alvari H, Hamzeh A, *et al.* Social network community detection using the Shapley value [C]// Proc of the 16th CSI International Symposium on Artificial Intelligence and Signal Processing. Washington DC: IEEE Computer Society, 2012: 222-228.

[25] Alvari H, Hashemi S, Hamzeh A. Discovering overlapping communities in social networks: A novel game-theoretic approach [J]. AI Communications, 2013, 26 (2): 161.

[26] Pizzuti C. Overlapped community detection in complex networks [C]// Proc of the 11th Annual conference on Genetic and Evolutionary Computation. New York: ACM Press, 2009: 859-866.

[27] Zhou Xu, Liu Yanheng, Zhang Jindong, *et al.* An ant colony based algorithm

for overlapping community detection in complex networks [J]. Physica A: Statistical Mechanics and its Applications, 2015, 427 (11): 289.

[28] Rees B S, Gallagher K B. Overlapping community detection using a community optimized graph swarm [J]. Social Network Analysis and Mining, 2012, 2 (4): 405.

[29] Wang Xiaohua, Jiao Licheng, Wu Jianshe. Adjusting from disjoint to overlapping community detection of complex networks [J]. Physica A: Statistical Mechanics and its Applications, 2009, 24 (24): 5045.

[30] Tanmoy C. Leveraging disjoint communities for detecting overlapping community structure [J]. Journal of Statistical Mechanics: Theory and Experiment, 2015, 2015 (5): 05017

[31] Tanmoy C, Srinivasan S, Ganguly N. On the Permanence of Vertices in Network Communities [C]// Proc of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2014: 1396-1405.

[32] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical review E, 2004, 69 (22): 026113.

[33] Chen Mingming, Kuzmin K, Szymanski B K. Extension of modularity density for overlapping community structure [C]// Proc of IEEE//ACM International Conference on Advances in Social Networks Analysis and Mining. Washington DC: IEEE Computer Society, 2014: 856-863.

[34] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms [J]. Physical Review E, 2008, 78 (4): 046110.

[35] Guimera R, Danon L, Diaz-Guilera A, *et al.* Self-similar community structure in a network of human interactions [J]. Physical Review E, 2003, 68 (62): 651031.

[36] Watts D J, Strogatz S H. Collective dynamics of'small-world'networks [J]. Nature, 1998, 393 (6684): 440.

[37] Boguñá M, Pastor-Satorras R, Díaz-Guilera A, *et al.* Models of social networks based on social distance attachment [J]. Physical Review E, 2004, 70 (52): 056122.